CS189: Introduction to Machine Learning

Homework 4

Due: March 12, 2013 @ 11:59PM

Submission: **bSpace/Kaggle**

Problem: Logistic Regression with Newton's Method

Let $\{(x_i, y_i)\}_{i=1}^n$ be a training set, where $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$. Recall the negative log likelihood for l_2-regularized logistic regression:

$$l(\beta) = \lambda \|\beta\|_2^2 - \sum_{i=1}^{n} [y_i \log \mu_i + (1 - y_i) \log(1 - \mu_i)]$$

where $\mu_i = 1/(1 + \exp(-\beta^T x_i))$, and $\lambda > 0$ is the regularization parameter.

In this problem, you will use Newton's method to minimize this negative log likelihood on a small training set. Here's the setup: We have four data points (in $\mathbb{R}^2$), two of class 1, and two of class 0. Here is the data (you may want to draw this on paper to see what the data looks like):

$$X = \begin{bmatrix} 0 & 3 \\ 1 & 3 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Here, X is the design matrix; each row x_i^T of X is a data point.

Notice that this data cannot be separated by a boundary that goes through the origin. To account for this, you should append 1 to the x_i vectors and fit a three-dimensional β vector that includes an offset term.

1. Derive the gradient of the negative log likelihood. Your answer should be a simple matrix-vector expression. Do NOT write your answer in terms of the individual elements of the gradient vector.

2. State the Hessian of the negative log likelihood. Again, your answer should be a simple matrix-vector expression.

3. State the update equation for Newton's method for this problem.

4. We are given that $\lambda = 0.07, \beta^{(0)} = \begin{bmatrix} -2 & 1 & 0 \end{bmatrix}^T$.

   (a) State the value of $\mu^{(0)}$ (the value of μ before any iterations).

(b) State the value of $\beta^{(1)}$ (the value of β after one iteration).

(c) State the value of $\mu^{(1)}$.

(d) After performing a second iteration, state the value of $\beta^{(2)}$.

Problem: Spam classification using Logistic Regression

The spam dataset given to you as part of the homework in `spam.mat` consists of 4601 email messages, from which 57 features have been extracted as follows:

- 48 features giving the percentage (0 - 100) of words in a given message which match a given word on the list. The list contains words such as business, free, george, etc. (The data was collected by George Forman, so his name occurs quite a lot!)
- 6 features giving the percentage (0 - 100) of characters in the email that match a given character on the list. The characters are ;([! $ # .
- Feature 55: The average length of an uninterrupted sequence of capital letters
- Feature 56: The length of the longest uninterrupted sequence of capital letters
- Feature 57: The sum of the lengths of uninterrupted sequence of capital letters

The dataset consists of a training set size 3450 and a test set of size 1151. One can imagine performing several kinds of preprocessing to this data. Try each of the following separately:

i) Standardize the columns so they all have mean 0 and unit variance.

ii) Transform the features using $log(x_{ij} + 0.1)$.

iii) Binarize the features using $\mathbb{I}(x_{ij} > 0)$.

Note that we haven't provided you with test labels for this homework. This means you won't be able to verify the test performance of your classifier like you've done before. Instead, we'll be using Kaggle, but more on that later.

For this homework, you need to do the following:

1. Derive the batch gradient descent equations for logistic regression with l_2 regularization and write them down (you can just state it if your derivation is in the previous problem).

   Choose a reasonable regularization parameter value, and plot the training loss (the negative log likelihood of the training set) vs the number of iterations. You should have one plot for each preprocessing method.

   Note: One iteration here amounts to scanning through the whole training data and computing the full gradient.

2. Derive stochastic gradient descent equations for l_2 regularized logistic regression. Plot the training loss vs number of iterations (again, you should have one plot for each preprocessing method). Do you see any differences from the corresponding curve from (1)? If so, why?

   Note: One iteration here corresponds to processing just one data point.

3. Instead of a constant learning rate (η), repeat (2) where the learning rate decreases as $\eta \propto 1/t$ for the t^{th} iteration. Plot the training loss vs number of iterations. Is this strategy better than having a constant η? Can you think any other strategies that might work well?

4. Now, tune your classfier choosing the most appropriate of the 3 preprocessing methods and tuning the regularization parameter. Submit your results to **Kaggle** (`http://tinyurl.com/cs189-kaggle`). Your classifier, when given the test points, should output a CSV file (there is a sample one on Kaggle). You'll upload this CSV file to Kaggle where it'll be scored with both a public test set, and a private test set. You will be able to see only your public score.

   You're only allowed **ONE SUBMISSION PER DAY**, so make sure you're confident in your submission. Since you won't have the test labels, the only way to test your classifier will be via validation. In your writeup, describe the process you used to decide which parameters to use for your best classifier.

   Beware of overfitting!

NOTE: You are NOT supposed to use any kind of software package for logistic regression!

Submission Instructions

In your submission, you need to include a write up with answers to all the questions and the plots. You also need to include your code a README with instructions as to how we can run your code.

Also, you should have at least one submission to Kaggle by the assignment due date.